Technical white paper

Migrating Symantec Veritas Cluster Server to HP Serviceguard for Linux

Table of contents

Executive summary

This white paper describes a way to migrate Symantec Veritas Cluster Server (VCS) setup to HP Serviceguard for Linux (SG/LX) setup. It provides an overview of functionalities and features of VCS and SG/LX, and maps VCS paradigm to SG/LX paradigm. It also includes a step-by-step process to migrate an existing VCS 6.0 setup to a SG/LX A.11.20.00 setup.

Introduction

This paper helps users to migrate from a Veritas Cluster setup to a SG/LX setup. HP Serviceguard has been one of the preferred HA solutions of the industry for over 20 years and is available on HP-UX Integrity and Linux x86_64 bit platforms. HP Serviceguard solutions for Linux are built to address the financial, business and operational impacts of downtime in a mission-critical x86 infrastructure. You can protect your mission critical workloads running on x86_64 bit platforms from downtime using SG/LX. It is a reliable, cost-effective-proven high availability (HA) solution that is easy to deploy and manage. It is fully automated and ready to support your multi-platform environment and diverse workloads. It helps maximize uptime by moving workloads between servers in the event of a failure, so quickly and transparently that users may not even realize it happened.

The first half of this paper provides VCS overview, SG/LX overview, mapping of a VCS cluster to an SG/LX cluster, mapping of a VCS system to an SG/LX node, mapping of a VCS service group to an SG/LX package. The second half of this paper describes a step-by-step procedure to migrate a VCS 6.0 cluster to an SG/LX A.11.20.00 cluster by using a sample example.

Audience

This document is targeted for the users of Symantec VCS who want to migrate to HP SG/LX. It is assumed that the reader has a general understanding of Symantec VCS and HP SG/LX. For more information on each solution, see hp.com/go/sglx and “Veritas™ Cluster Server Administrator’s Guide”.

Converting a VCS setup to SG/LX setup

SG/LX overview

HP Serviceguard solutions for Linux A.11.20.00 provides critical applications with the HA that enterprise customers require for 24x7 business operations on an x86 platform. SG/LX allows users to create HA clusters of HP ProLiant servers that are running Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server. HP Serviceguard solutions for Linux HA clusters are designed to protect applications from a wide variety of software and hardware failures by monitoring the health of each server (node) and quickly responding to failures of system processes, system memory, LAN media and adapters, or applications. In the event when one component fails, the redundant component takes over without compromising data integrity. HP SG/LX on its own does not require planned downtime for maintenance. It can also help you perform application, OS, and hardware maintenance operations effectively against the ever shrinking planned downtimes available for such maintenance operations.

HP provides multiple extensions and toolkits that provide out-of-the-box deployment of popular mission critical applications in a SG/LX cluster. The SG/LX portfolio also contains other value-added solutions that provide disaster recovery capabilities by leveraging robust Serviceguard automatic failover with data redundancy technologies.

- **Serviceguard Manager:** Serviceguard Manager is a GUI to administer SG/LX cluster.
- **Extended Distance Cluster (XDC):** The XDC protects against multiple failures by extending a single SG/LX cluster across two data centers separated up to 100 km distance. It uses host based data replication.
- **Toolkits:** HP Serviceguard provides toolkits to manage popular applications, for example, Oracle, NFS, SAP, Apache, and others.
- **HP Metrocluster for Linux:** HP Metrocluster is a disaster recovery solution that helps ensure service availability in the event of disaster. It enables you to create a single cluster using nodes that are located apart from each other at different data centers over metropolitan distance. This solution integrates SG/LX robust failover mechanism with disk array based replication technologies. This allows applications to automatically failover across data centers and use disk array based replication between the data centers. The SG/LX failover can automatically manage the replication preserving data integrity and continuing protection against future failures. Metrocluster for Linux is available for the following HP array based replication technologies: HP 3PAR P10000/P7000 Remote Copy, HP XP P9000 Continuous Access, and EVA P6000 Continuous Access.

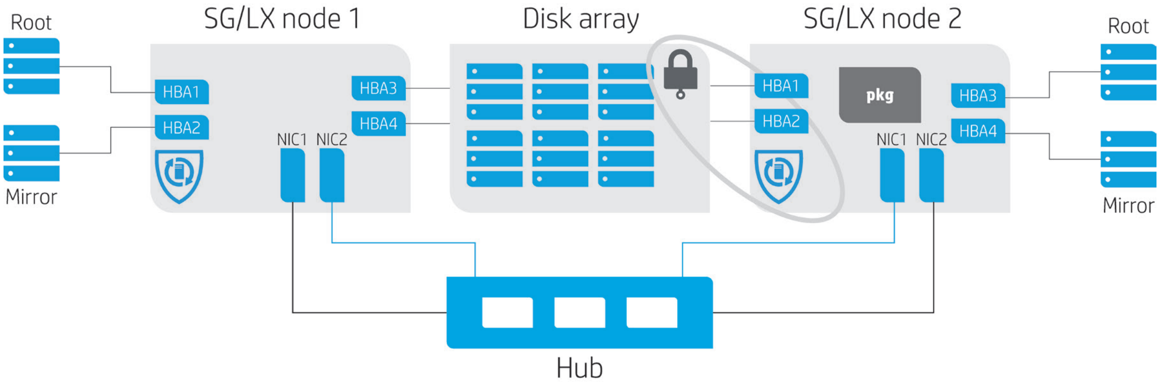
- **HP Continentalclusters for Linux:** HP Continentalclusters protects and manages multiple clusters where the clusters are located apart from each other over any distance. With Continentalclusters one can perform a push-button recovery on the recovery site and use any type of replication to replicate data between the data centers. HP provides preintegrated solutions with Continentalclusters and disk array based replication for the following arrays like HP 3PAR P10000/P7000 Remote Copy, HP XP P9000 Continuous Access, and EVA P6000 Continuous Access. In addition, you can integrate any host based replication technology with Continentalclusters using the data-sender and data-receiver framework in Continentalclusters. You can also use Continentalclusters with other non HP disk-array based replication. In such cases, Continentalclusters can be used to automate the recovery process, and the user is responsible for data currency and consistency before initiating a recovery.

SG/LX cluster membership

An SG/LX cluster is a networked grouping of HP ProLiant systems (host systems known as nodes) having sufficient redundancy of software and hardware that a single failure may not significantly disrupt service. Application services (individual Linux processes) and their corresponding storage and network resources are deployed together using the package construct. In the event of a single service, node, network, or other resource failure, SG/LX can automatically transfer control of the package to another node within the cluster, allowing services to remain available with minimal interruption.

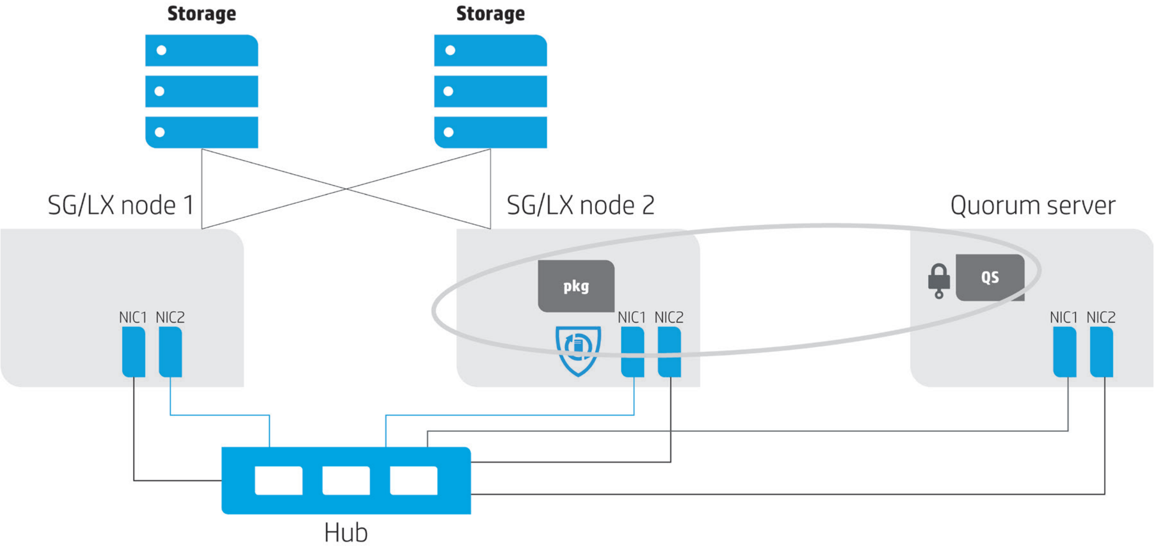
- **Cluster quorum**
  In the case of a failure, SG/LX cluster reforms with the surviving nodes. In SG/LX each cluster node has one vote. During cluster reformation, the nodes that can communicate with each other, regroup and attempt to reform a cluster. In SG/LX, a cluster can be reformed only if the cluster quorum condition is achieved. The cluster quorum condition is met only when a strict majority (that is, more than 50%) of the nodes running previously is available to form the cluster. All and any remaining nodes that are not in the group with cluster quorum condition can be reset. Since the quorum is dynamic (The required quorum is generally based on the previously running cluster membership and not on the static count of the number of nodes configured in the cluster.), SG/LX clusters can reform even after repeated multiple cluster node failures.
- **Split brain**
  If a failure in the SG/LX cluster results in equal halves (exactly 50%) of a previously running cluster, then this situation is called as split brain. A robust arbitration method is used to break a tie. This robust arbitration method guarantees that only one half can form a cluster and the other half may be forced to shut down. In SG/LX, arbitration can be implemented by cluster lock. Either Lock logical unit number (LUN) or Quorum Server can be configured as cluster lock. Configuring a membership arbitration method (cluster lock) is a must in the clusters that have two nodes.
- **Lock LUN**
  Lock LUN acts as a quorum arbitration method in the SG/LX clusters. It is a special area in the storage which is sharable by all nodes in the cluster. During split brain situation, each partition races with the other to obtain a lock on the Lock LUN. Once the lock is obtained, it is marked so that the other partition can recognize the Lock LUN as "taken". Serviceguard periodically checks the status of Lock LUN device and logs messages in the syslog in case of a failure. The Lock LUN is dedicated to use as only membership arbitration method, which is also called as cluster lock. HP recommends that the Lock LUN should comprise of whole disk. The Lock LUN can be used only in the cluster which has nodes up to four. The following diagram shows Lock LUN configuration in a 2-node cluster. When there is loss of communication between node 1 and node 2, both nodes race to acquire the lock and node 2 wins the race by acquiring the lock. The node 2 is allowed to continue running and node 1 is reset.

Figure 1. 2-node SG/LX cluster using Lock LUN as a cluster lock



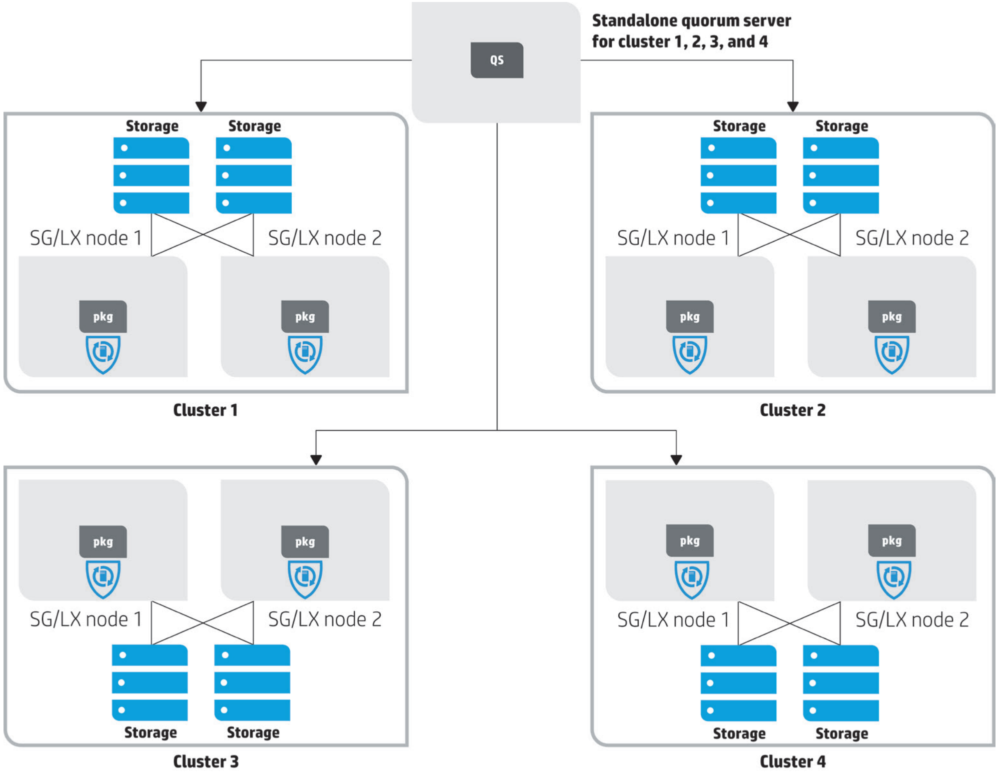
- **Quorum Server software**
  Quorum Server can be used as a membership arbitration method for multiple clusters at a time. The Quorum Server does not have any limitation in terms of number of nodes in a cluster. The Quorum Server can be configured as a Serviceguard package or standalone, but in either case it must run on a system outside of the cluster for which it is providing quorum services. The Quorum Server listens to connection requests from the Serviceguard nodes on a known port. The server maintains a special area in memory for each cluster, and when a node obtains the cluster lock, this area is marked so that other nodes can recognize the lock as "taken". The operation of the Quorum Server in a 2-node cluster is shown in the following diagram (figure 2). When there is a loss of communication between node 1 and node 2, the Quorum Server chooses one node (in this example, node 2) to continue running in the cluster. The other node (node 1) is reset.

Figure 2. 2-node SG/LX cluster using Quorum Server as a cluster lock



The following diagram shows a Quorum Server providing quorum service to multiple clusters.

Figure 3. Single Quorum Server providing quorum service to multiple clusters



The Quorum Server software is available for HP-UX 11i v2, v3, RHEL, and SUSE Linux Enterprise Server. The software can be downloaded free from software.hp.com. A single Quorum Server can serve multiple SG/LX and Serviceguard for HP-UX clusters. Ideally, the Quorum Server must be configured in a separate service cluster for HA in fault domain different from the fault domain of servers in the cluster serviced by them.

- **Node fencing**
  A reboot is done if a cluster node cannot communicate with the majority of cluster members for a predetermined time, split brain or under other circumstances such as a kernel hang or failure of the cluster daemon (cmcld). This reboot is initiated by DEADMAN driver, which act as a robust fencing mechanism in SG/LX cluster. There is no manual configuration needed to configure DEADMAN driver. The DEADMAN driver is a dynamically loadable kernel module that is compiled into the kernel automatically when SG/LX is installed, but if you are planning to update Linux kernel, then it is required to recompile the kernel with DEADMAN driver.

Networking in the SG/LX cluster

SG/LX uses one or more heartbeat networks to send heartbeat among all the nodes and to maintain the cluster membership. SG/LX also uses the heartbeat network for communication between nodes. To build a resilient cluster it is crucial to have resilient heartbeat network infrastructure. So it is recommended to configure multiple heartbeat networks and use channel bonding technologies at link level.

Applications deployed in SG/LX cluster can use their own network in the cluster, for their client access. Such networks are called client access networks. SG/LX can monitor and manage IP addresses used by applications on these networks. During failover, SG/LX can move the application’s IP address on the client access network from the failed node to the target node. Also, it is okay to configure the client access networks for heartbeat network traffic. SG/LX exchanges only small messages and does not have demanding bandwidth requirements. So using client access networks for heartbeat does not affect application accessibility or performance.

VCS overview

VCS is a HA solution from Symantec VCS (VCS) connects multiple, independent systems into a management framework for increased availability. Each system or node runs its own OS and cooperates at the software level to form a cluster. VCS links commodity hardware with software to provide application failover and control. When a node or a monitored application fails, other nodes can take predefined actions to take over and bring up the failed application elsewhere in the cluster.

VCS cluster membership

VCS cluster uses two types of communications, intra system communications and inter system communications. In the intra system communications, VCS uses protocol called “Inter Process Messaging (IPM)” to communicate with the GUI, the command line and the agents. In the inter system communications, VCS uses the cluster interconnects for network communication between cluster systems. The Group Membership Services/Atomic Broadcast protocol (GAB) is responsible for cluster membership and reliable cluster communications.

When a system crashes or is powered off, it stops sending heartbeats to other systems in the cluster. By default, other systems in the cluster wait 21 seconds before declaring it dead. The time of 21 seconds derives from 16 seconds default timeout value for Low Latency Transport (LLT) peer inactive timeout (can be modified by users), plus 5 seconds default value for GAB stable timeout (can be altered by users).

Each system in the cluster runs a kernel fencing module which is responsible for ensuring valid and current cluster membership during a membership change through the process of membership arbitration. VCS cluster uses a special service called the coordination point for membership arbitration. Coordination points provide a lock mechanism to determine which nodes get to fence off data drives from other nodes. A node must eject a peer from the coordination point before it can fence the peer from the data drive. The coordination point can be a disk or a server. VCS recommends configuring three coordinator points. The kernel fencing module registers with the coordination points during normal operation. At the time of cluster reformation, the kernel fencing module of each system races for the control of the configured coordination points. VCS prevents split brain by allowing the winner partition to fence the ejected nodes from accessing the data disks.

Comparison of some important terminologies

The following table compares the important VCS terminologies with SG/LX terminologies.

Table 1. Important terminologies

	VCS terminology	SG/LX terminology	Descriptions
1	System	Node	Member of cluster
2	Service group	Package	Way to package all hardware and software resource together as single unit
3	Resource agent	Modules	Way to manage (start/stop/monitor) hardware and software resources via SG/LX package or VCS service groups
4	Agent	Toolkit	A framework to manage (start/stop/monitor) a specific application
5	Disk fencing	Lock LUN	One of the cluster membership arbitration mechanisms
6	Coordination point server	Quorum Server	One of the cluster membership arbitration mechanisms
7	VCS Operation Manager	Serviceguard Manager	GUI
8	Failover service groups	Failover package	A packaged single unit which contains software and hardware resources that run on one node/system at a time
9	Parallel service groups	Multi node package	A service group/package that can run on multiple nodes at a time
10	Global clusters	Continentalclusters for Linux	Disaster recovery solution in which multiple clusters are used to provide application recovery over LAN or WAN
11	Campus or metropolitan clusters	XDC and Metrocluster for Linux	A single cluster that spans across multiple data centers or buildings
12	Cluster	Cluster	Group of servers

Mapping of VCS cluster attributes to SG/LX cluster attributes

The following table maps the VCS cluster attributes to the closest SG/LX cluster attributes.

Table 2. VCS cluster attributes and SG/LX cluster attributes

	VCS cluster attributes, which can be configured by users	Corresponding SG/LX cluster attributes (equivalent to some extent), which can be configured by users	Descriptions
1	AdministratorGroups, Administrator	user_name, user_host, and user_role	In VCS, the “AdministratorGroups” and “Administrator” attributes are used to specify list of OS user account groups or list of OS users who have administrative privileges on the cluster. This attribute applies to clusters running in secure mode. Serviceguard provides role-based access control (RBAC) which is used to grant Administrator privilege to multiple OS users by specifying USER_ROLE as “FULL_ADMIN”.
3	ClusterName	cluster_name	The name of the cluster
4	EnableFFDC	cmcluster.conf file	In VCS, you can enable or disable Fast failure data capture (FFDC) logging using “EnableFFDC” attribute. By default, FFDC logging is enabled. In Serviceguard, the “/etc/cmcluster.conf” file is used to enable logging of various daemons.
5	EnginePriority	No attribute	In VCS, you can define the priority in which high-availability daemon (HAD) runs by using the “EnginePriority” attribute. In Serviceguard, core daemons usually run at highest priority.

	VCS cluster attributes, which can be configured by users	Corresponding SG/LX cluster attributes (equivalent to some extent), which can be configured by users	Descriptions
6	Guests	user_name, user_host, and user_role	In VCS, the "Guests" attribute is used to specify a list of OS user accounts that have Guest privileges on the cluster. Serviceguard provides RBAC to specify Guest privilege to multiple OS users. To achieve this, USER_ROLE is specified as "MONITOR".
7	GroupLimit	max_configured_packages	In VCS, the "GroupLimit" attribute is used to specify maximum number of service groups. In Serviceguard, the "max_configured_packages" attribute is used to specify the maximum number of packages.
8	LockMemory	No attribute	In VCS, the "LockMemory" attribute is used to control the locking of VCS engine pages in memory. Serviceguard core daemons are pinned in the memory by default.
9	LogSize	No attribute	In VCS, the "LogSize" indicates the size of engine log files in bytes.
10	OperatorGroups and Operators	user_name, user_host, and user_role	In VCS, the "OperatorGroups" and "Operator" attributes allow you to specify list of OS user groups or users who have Operator privileges on the cluster. In Serviceguard, cluster and package access control policy parameters can be used together to get the different level of privileges for multiple users. The user_role parameter determines the role of the user. Following are the supported values for user_role attribute: • MONITOR: read-only capabilities for cluster and packages • PACKAGE_ADMIN: MONITOR, plus administrative privileges for packages • FULL_ADMIN: MONITOR and PACKAGE_ADMIN plus the administrative privileges for the cluster.
11	PanicOnNoMem	No attribute	In VCS, "PanicOnNoMem" indicates the action that you want VCS engine (HAD) to take if it cannot receive messages from GAB due to low-memory. Serviceguard usually resets the system when sufficient memory is not allocatable.
12	UseFence	forced_pr_disable	In VCS, the "UseFence" attributes to enable or disable SCSI-3 persistent reservation (PR) based I/O fencing. In Serviceguard, SCSI-3 PR based I/O fencing is enabled by default. To disable SCSI-3 PR based I/O fencing for all packages in the SG/LX cluster set the parameter forced_pr_disable in the cluster configuration file to 1.

Note:
HacliUserLevel, HostMonLogLvl, LogClusterUUID, PrintMsg, ProcessClass, ProcessPriority, SourceFile, Stewards, TypeLimit, UserNames, EngineShutdown, DeleteOnlineResource, CounterMissTolerance, EngineClass, CounterMissAction, and CounterInterval are not relevant in SG/LX Environment.

Mapping of VCS system attributes to SG/LX node attributes

The following table maps the VCS system attributes to the closest SG/LX node attributes.

Table 3. VCS system attributes and SG/LX node attributes

	VCS system attributes, which can be editable by users	Corresponding SG/LX node attributes (equivalent to some extent), which can be editable by users	Descriptions
1	Capacity	"capacity_name" and "capacity_value"	In VCS, the "Capacity" attribute is configured to express total system load capacity. This value is relative to other systems in the cluster and does not reflect any real value associated with a particular system. In Serviceguard, "capacity_name" and "capacity_value" attributes are used to define capacities of nodes.
2	DynamicLoad	Generic Resource framework	In VCS, the "DynamicLoad" attribute is used to indicate system-maintained value of current dynamic load. The value is set external to VCS with the hasys-load command. In Serviceguard, Generic Resource framework is used for modeling dynamic metrics to influence failover.
3	SourceFile	"cmclconfig" file	In VCS, the "SourceFile" attribute indicates a file from which the configuration is read. In Serviceguard, configuration file is stored in binary format with the name "cmclconfig".
4	SystemLocation	SITE_NAME	In VCS, the "SystemLocation" attribute can be used to indicate the location of the system. In Serviceguard, SITE_NAME attribute is used to indicate the location of nodes.
5	UserInt	PEV_	In VCS, the attribute "UserInt" is used to store integer values. VCS does not interpret the value of this attribute. In Serviceguard, user-defined variables can specify using PEV_ prefix in the application's package configuration file.

Converting VCS service groups to SG/LX packages

The VCS service groups and the SG/LX packages

VCS service group acts as a virtual container that contains all the hardware and software resources that are required to run an application. The VCS service groups allow VCS to control all the hardware and software resources of the managed application as a single unit.

The SG/LX package can be defined as a collection of services, disk volumes, generic resources, and IP addresses that are managed by Serviceguard to help ensure they are available. In case of any hardware or software failure, which impacts any component of the SG/LX package, Serviceguard fails over the package to another node in the cluster. This allows the application services to continue with minimal interruption.

VCS service groups and agents overview

VCS service groups

VCS service group can be defined as a set of resources working together to provide application services to clients. It can be made dependent on or independent of another service group. In the event of a failure, the entire service group fails over without affecting others if that service group does not have any dependency on another. The service group dependencies can be used to create more complex managed applications. A single node can host any number of service groups. It can be classified into three groups: failover, parallel, and hybrid.

- **Failover group:** A failover service group runs on one system in the cluster at a time. Failover groups are used for most applications that do not support multiple systems to simultaneously access the application's data.
- **Parallel group:** A parallel service group is more complex than a failover group. Parallel service groups are appropriate for the applications that manage multiple application instances that run simultaneously without data corruption.
- **Hybrid group:** A hybrid service group is for the replicated data clusters and is a combination of the failover and parallel service groups. It behaves as a failover group within a system zone and a parallel group across system zones.

VCS agents

VCS agents are multi-threaded processes that manage resources. VCS has one resource agent per resource type. A single agent manages all resources of that type. The agent starts the resource, stops the resource, periodically monitors the resource, and updates the VCS engine with the resource status. The VCS agents fall into three categories:

- **Bundled agents:** These agents are packaged with VCS. These include agents for disk, IP, mount, etc.
- **Enterprise agents:** The enterprise agents control third party applications. These include agents for Oracle, Sybase, and Database 2 (DB2).
- **Custom agents:** Typically, custom agents are developed because the user requires control of an application that the current bundled or enterprise agents do not support. Usually, customers or Symantec consultants develops the custom agents.

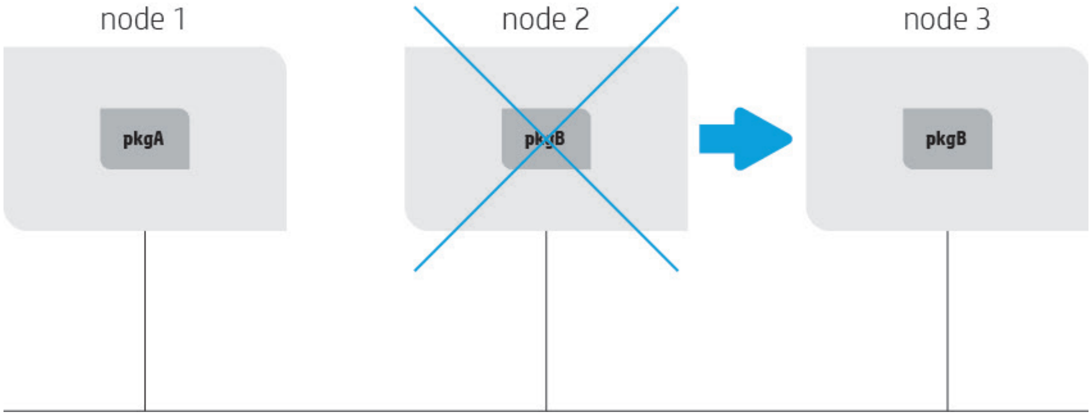
SG/LX packages and toolkits overview

SG/LX packages

The SG/LX packages are the means by which Serviceguard starts and halts configured applications. An SG/LX package can be treated as a single unit of failover in the SG/LX cluster. The SG/LX package can be classified into three groups:

- **Failover packages:** A failover package runs on one node at a time. In case of a service, network, or other resource or dependency failure, the package failover takes place. A package failover involves both halting the existing package and starting the new instance of the package on a new node.

Figure 4. SG/LX package moving during failover



- **Multi node packages:** A multi node package can run on multiple nodes simultaneously. It can be independently halted or started on the individual nodes. Failures of package components such as applications, services, generic resource, or subnets may cause the package to be halted only on the node on which the failure occurred.
- **System multi node packages:** System multi node packages are supported only for the applications supplied by HP. A system multi node package runs on all cluster nodes at the same time. It cannot be started and halted on the individual nodes.

SG/LX toolkits

The SG/LX toolkits enable easy integration of applications with SG/LX to provide HA. SG/LX provides separate toolkits for supported applications to create application specific Serviceguard package. The SG/LX toolkit starts, monitors, and stops the application. SG/LX provides the following toolkits:

- **Oracle database (DB) toolkit:** The Serviceguard Oracle DB toolkit manages Oracle DB. It starts, monitors, and stops Oracle DB instance. This toolkit is priced.
- **NFS toolkit:** The Serviceguard NFS toolkit manages NFS server. It starts, monitors, and stops the NFS Server. This toolkit can be downloaded for **free** from software.hp.com.
- **SGeSAP toolkit:** The Serviceguard extension for SAP toolkit manages SAP instance. It starts, monitors, and stops the SAP instance. This toolkit is priced.
- **HP Serviceguard Contributed Toolkits:** The Contributed Toolkit is a collection of toolkits that integrate popular applications with SG/LX. Following are the names of Contributed Toolkits. These toolkits can be downloaded for **free** from software.hp.com.
  - Apache toolkit
  - MySQL toolkit
  - PostgresSQL toolkit
  - Samba toolkit
  - Sendmail toolkit
  - Tomcat toolkit
- **Serviceguard Developer's Toolbox:** Serviceguard Developer's Toolbox enables users to write a toolkit themselves. Typically, users write toolkits using Serviceguard Developer's Toolbox for that application whose toolkit is not available. The Developer's Toolbox can also be downloaded from software.hp.com for **free**.

Mapping VCS service group attributes to SG/LX package attributes

This section compares VCS service group with SG/LX package and describes the possible mapping of attributes.

VCS enables users to specify all hardware and software resources in a single service group. VCS defines resource type for each resource it manages. For example, the IP resource type can be used to manage IP address resource. VCS provides corresponding agent for each resource type. A single resource agent manages all resources of that type. For example, the IP resource agent manages all IP resources.

Like VCS, SG/LX also enables users to specify the hardware and software resources in a single unit. That single unit is called package in a SG/LX cluster. Users can specify services, disk volume, IP addresses, generic resources, etc., in a single package. SG/LX also provides separate script to manage each resource. These separate scripts are called modules, and these can be specified while creating a package. Examples of some modules are filesystem, volume group, and package_ip.

Mapping VCS service group dependency to Serviceguard package dependency

A VCS service group can be dependent on another service group. Same service group may act as both parent and child in a dependency tree. In VCS, a dependency tree may be up to five levels deep. The dependency relationship between a parent and a child service group can be characterized by the dependency category, the location of the service groups, and the rigidity of dependency.

Similarly, an SG/LX package can be also dependent on another package. The same package can act as both parent and child in a dependency tree. The SG/LX dependency relationship between a parent and a child can be characterized by the dependency name, condition, and location of the package.

Following are the possible values for dependency categories in VCS:

- **Online group dependency:** The parent service group must wait for the child service group to be brought online before it can start.
- **Offline group dependency:** The parent service group can be started only if the child service group is offline and vice versa.

Following are the corresponding possible values for the dependency conditions in SG/LX:

- **up:** The dependent successor package must wait for the predecessor package to be brought online before it can start.
- **down:** The dependent successor package can be started only if the predecessor package is offline and vice versa.

Following are the possible values for dependency location in VCS:

- **Local dependency:** The parent service group depends on the child service group for being online or offline on the same system.
- **Global dependency:** An instance of the parent service group depends on one or more instances of the child service group being online on any system in the cluster.
- **Remote dependency:** An instance of the parent service group depends on one or more instances of the child service group being online on any system in the cluster other than the system on which the parent service group is online.

Following are the corresponding possible values for dependency location in SG/LX:

- **same_node:** It means that the dependency must be satisfied on the same node.
- **any_node:** It means that the dependency can be satisfied on any node in the cluster.
- **different_node:** It means that the dependency must be satisfied on a node other than the dependent package's node.

Following are the possible values for the dependency rigidity in VCS:

- **Soft dependency:** It specifies the minimum constraints while bringing parent and child groups online.
- **Firm dependency:** It imposes more constraints when VCS brings the parent or child groups online or takes them offline.
- **Hard dependency:** It imposes the maximum constraints when VCS brings the parent of child service groups online or takes them offline.

There is no configurable parameter in SG/LX cluster to define the rigidity of a dependency. The VCS firm dependency rigidity is the closest equivalent for SG/LX package dependency rigidity.

Mapping VCS resources to Serviceguard modules

In VCS, you can include resources in a service group and use corresponding resource agent for monitoring. For example IP resource agent, filesystem resource agent, mount resource agent, and others. You can define the order in which the resources can start and stop by defining dependency among resources.

In Serviceguard, you can also include resources in a package by including corresponding resource module. For example, modules of volume group, IP, filesystem, and others. These modules only start and stop the resources. For monitoring the resources, you can configure generic resources in the SG/LX package. To configure a generic resource, a monitoring script needs to be written which can monitor the resource and set the status of generic resource. SG/LX recommends launching the monitoring scripts by configuring them as services in a package.

Mapping VCS user privileges and roles to SG/LX access control policy parameters

Cluster operations are enabled or restricted depending on the privileges with which you log on. VCS has three privilege levels: Administrator, Operator, and Guest. VCS provides some predefined user roles; each role has specific privilege levels. For example, the role Guest has the fewest privileges and the role Cluster Administrator has the most privileges.

In SG/LX, you can also enable or restrict the uses by defining access control policy parameters. Following three parameters are used to achieve this:

- **user_name:** It can either be "any_user" or a maximum of eight login names from the/etc/passwd file on user host.
- **user_host:** It is where the user can issue Serviceguard commands. It can be "any_serviceguard_node", or (any) "cluster_member_node", or a specific node.
- **user_role:** It can have one of the following three values:
  - MONITOR: read-only capabilities for the cluster and packages
  - PACKAGE_ADMIN: MONITOR, plus administrative privileges for packages in the cluster
  - FULL_ADMIN: MONITOR and PACKAGE_ADMIN plus the administrative privileges for the cluster

Mapping VCS system capacity and service group load to Serviceguard node capacity and package weight

The VCS capacity construct enables you to define a fixed amount of resources a server can provide. It sets a fixed load-handling capacity for servers. The Load attribute allows you to define a fixed amount of resources a specific service group is expected to utilize (Load). It sets a fixed demand for service groups. When a service group is brought online, its load is subtracted from the system's capacity to determine available capacity. VCS maintains this info in the attribute Available Capacity. When a failover occurs, VCS determines which system has the highest available capacity and starts the service group on that system.

SG/LX also enables you to define the capacities for a node and corresponding weight for a package. The capacity of a node determines how much weight of packages it can accommodate. For example, you can define a capacity called memory, which has value 1000 for a node. This means that node can accommodate a package or multiple packages having weight less than or equal to 1000. If you have defined package A with weight 1000, package B with weight 500, and package C with weight 500, then that node can either accommodate only package A, or both package B and package C. In SG/LX, you can use following parameters to define capacity of nodes and weight of packages. You can define multiple capacities and weights if required.

- **capacity_name:** It provides the name of the capacity. This is a node attribute.
- **capacity_value:** It provides the value of the capacity. This is a node attribute.
- **weight_name:** It provides the name of the weight. It should be the same as one of the capacity names. This is a package attribute.
- **weight_value:** It provides the value of the weight. This attribute is associated with the package. The value of weight gets subtracted from the node's capacity value where it is running.

SG/LX uses first fit policy when capacities and weights are configured. Based on the failover policy of a package, SG/LX creates a list of adoptive nodes for failing over that package. If capacities and weight are configured, SG/LX chooses the first adoptive node which can accommodate the package weight for failover.

Table 4. VCS service group attributes and SG/LX package attributes

	VCS service group attributes, which can be editable by users	**Corresponding SG/LX package attributes (equivalent to some extent) present in package ASCII file, which can be editable by users**	**Descriptions**
1	AdministratorGroups, Administrators	user_name, user_host, and user_role	In VCS, the "AdministratorGroups" and "Administrators" attributes are used to provide administrative privileges on the service group to list of OS user account groups and users. Serviceguard provides access control policy parameters to provide Administrator access to multiple OS users. Administrator access to multiple users is provided by listing multiple users with "FULL_ADMIN" as USER_ROLE.
2	AutoFailOver	auto_run	In VCS, the "AutoFailOver" attribute indicates whether VCS initiates an automatic failover if the service group faults. In Serviceguard, the "auto_run" attribute is enabled to initiate an automatic failover of a package if that package fails.
3	AutoRestart	service_restart	In VCS, the "AutoRestart" attribute is used to restart a service group after a faulted persistent resource becomes online. In Serviceguard, the "service_restart" attribute is used to determine whether a service of package can be restarted or not.
4	AutoStart	auto_run	In VCS, the "AutoStart" attribute designates whether a service group is automatically started when VCS is started. In Serviceguard, the "auto_run" attribute determines whether a service_group is automatically started when VCS is started or not.
5	AutoStartList	node_name	In VCS, the "AutoStartList" attribute is used to specify the list of systems on which, under specific conditions, the service group can be started with VCS (usually at system boot). In Serviceguard, the "node_name" attribute is used to specify the order of nodes in which package can start.

	VCS service group attributes, which can be editable by users	**Corresponding SG/LX package attributes (equivalent to some extent) present in package ASCII file, which can be editable by users**	**Descriptions**
6	AutoStartPolicy	failover_policy	In VCS, the "AutoStartPolicy" attribute sets the policy which VCS uses to select the system to bring a service group online if multiple systems are available. This attribute has three options: Order, Load and Priority. In Serviceguard, the following failover policies can be used to determine which node should be chosen to bring up a package: • "configured_node": ordered list of nodes • "min_package_node": A node, which has minimum number of packages, can be selected to bring up the package or failover the package.
7	ClusterFailOverPolicy	failover_policy	In VCS, the "ClusterFailOverPolicy" attribute determines how a global service group behaves when a cluster faults or when a global group faults. The attribute can take the following values: • **Manual:** The group does not fail over to another cluster automatically. • **Auto:** The group fails over to another cluster automatically if it is unable to failover within the local cluster or if the entire cluster faults. • **Connected:** The group fails over automatically to another cluster only if it is unable to fail over within the local cluster. With HP Metrocluster, a cluster can be divided into sites. The sites can be located far from each other. Serviceguard provides different failover policies for the sites: • "site_preferred_manual": This failover_policy helps ensure that the SG/LX package never fails over to another site automatically. • "site_preferred": This failover_policy helps ensure that the SG/LX package fails over to another site only when it is unable to fail over within the current site.
8	Enabled	auto_run	In VCS, the "Enabled" attribute indicates if a service group can be failed over or brought online. In Serviceguard, the "auto_run" attribute determines if a package can be failed over or brought online.
9	Evacuate	cmhaltnode –f	In VCS, the "Evacuate" attribute initiates an automatic failover before stopping VCS on the node. In Serviceguard, the "cmhaltnode –f" command initiates automatic failover of the packages, provided the global and node switching is enabled for those packages. This is a default Serviceguard behavior.
10	FailOverPolicy	failover_policy	In VCS, the "FailOverPolicy" attribute sets the policy, which VCS uses, to select a system on which a service group will failover to if multiple systems exist. This attribute can take the following values: • **Priority:** The system defined as the lowest priority in the SystemList attribute is chosen. • **Load:** The system defined with the least value in the system's Load attribute is chosen. • **Round robin:** Systems are chosen according to how many active service groups they are hosting. The system with the least number of active service groups is chosen first. In Serviceguard, the "failover_policy" attribute determines which node be chosen to start or failover the package. Following values are used for "failover_policy" parameter in Serviceguard cluster: • "configured_node": The node which is first in the eligible nodes list is selected as an adoptive node. • "min_package_node": The node which is running the minimum number of packages is selected as an adoptive node.

	VCS service group attributes, which can be editable by users	**Corresponding SG/LX package attributes (equivalent to some extent) present in package ASCII file, which can be editable by users**	**Descriptions**
11	FaultPropagation	No attribute	In VCS, the “FaultPropagation” attribute specifies if VCS should propagate the fault up to parent resources and take the entire service group offline when a resource faults. In Serviceguard, failure of any configured resource in a package can cause the package to failover.
12	GroupOwner	email_id	In VCS, the “GroupOwner” attribute is used for VCS email notification and logging. VCS sends email notification to the person designated in this attribute when events occur that are related to the service group. Note that while VCS logs most events, not all events trigger notifications. In Serviceguard, the “Alert notification” feature is used to send mail to the configured email aliases on the package failure events.
13	GroupRecipients	email_id	In VCS, the “GroupRecipients” attribute is used for VCS email notification. VCS sends email notification to persons designated in this attribute when events related to the service group occur and when the event’s severity level is equal to or greater than the level specified in the attribute. In Serviceguard, the “Alert notification” feature is used to send mail to the configured emails on the package failure events.
14	Guests	user_name, user_host, and user_role	In VCS, the “Guests” attribute is used to provide Guest privileges to the list of OS user on the service group. In Serviceguard, access control policy parameters are used to grant read-only privilege to users by specifying “MONITOR” as USER_ROLE.
15	Load	weight_name and weight_value	In VCS, the “Load” attribute specifies an integer value expressing total system load the service group can put on a system. In Serviceguard, the “weight_name” and “weight_value” attributes are used to define the weight of an SG/LX package. The value of “weight_name” should be the same as one of the capacities defined in the cluster. The value of a package’s weight gets subtracted from the node’s capacity value where the package runs on that node.
16	ManageFaults	No attribute	In VCS, the “ManageFaults” attribute specifies if VCS manages resource failures within the service group by calling the clean function for the resources. In Serviceguard, on the failure of package, manual intervention is needed to do a proper cleanup.
17	ManualOps	No attribute	In VCS, the “ManualOps” attribute indicates if manual operations are allowed on the service group. Serviceguard does not recommend performing manual operation on packages.
18	OnlineRetryLimit	service_restart	In VCS, the “OnlineRetryLimit” attribute specifies the number of times a VCS engine tries to restart a faulted service group on the same system on which the service group faulted, before it gives up and tries to fail over the group to another system. In Serviceguard, the “service_restart” attribute specifies the number of time the service of the package can be restarted on a particular node.

	VCS service group attributes, which can be editable by users	Corresponding SG/LX package attributes (equivalent to some extent) present in package ASCII file, which can be editable by users	Descriptions
19	OperatorGroups, Operators	user_name, user_host, and user_role	In VCS, the “OperatorGroups” and “Operators” are used to specify a list of OS user groups and VCS users who can have Operator privilege on the service group. This attribute applies to clusters running in secure mode. In Serviceguard, cluster and package access control policy parameters are used to specify multiple users with different roles. The user_role parameter determines the role of the user. Following are the supported values for user_role attribute: • MONITOR: read-only capabilities for cluster and packages • PACKAGE_ADMIN: MONITOR, plus administrative privileges for packages • FULL_ADMIN: MONITOR and PACKAGE_ADMIN plus administrative privileges for the cluster.
20	Parallel	package_type	In VCS, the “Parallel” attribute indicates if service group is failover (0), parallel (1), or hybrid (2). In Serviceguard, the “package_type” attribute is used to determine the type of package. Following are the possible values of the “package_type” attribute: • failover • multi_node • system_multi_node
21	Priority	priority	In VCS, the “Priority” attribute enables users to designate and prioritize the service group. VCS does not interpret the value; rather, this attribute enables the user to configure the priority of a service group and the sequence of actions required in response to a particular event. In Serviceguard, the “priority” attribute is used to specify priority of a package. The lowest value of “priority” attribute means the highest priority.
22	SystemList	node_name	In VCS, the “SystemList” attribute is used to specify list of systems on which the service group is configured to run and their priorities. Lower numbers indicate a preference for the system as a failover target. In Serviceguard, multiple nodes can be specified as values for the “node_name” attribute. These multiple values serve as an ordered list, which is used to start or failover the package.
23	UserIntGlobal	pev_ prefix	VCS users can use the “UserIntGlobal” attribute for any purpose. It is not used by VCS. In Serviceguard, user-defined variables can be specified using pev_ prefix.
24	UserStrGlobal	pev_ prefix	VCS uses the “UserStrGlobal” attribute in the ClusterService group. Do not modify this attribute in the ClusterService group. Use the attribute for any purpose in other service groups. In Serviceguard, user-defined variables can be specified using pev_ prefix.
25	UserIntLocal	pev_ prefix	VCS users can use the “UserIntLocal” attribute for any purpose. It is not used by VCS. In Serviceguard, user-defined variables can be specified using pev_ prefix.
26	UserStrLocal	pev_ prefix	VCS users can use the “UserStrLocal” attribute for any purpose. It is not used by VCS. In Serviceguard, user defined variables can be specified using pev_ prefix.

NOTE:
AutoStartIfPartial, OnlineRetryInterval, Prerequisites, PreSwitch, SysDownPolicy, PrintTree, and SystemZones are not relevant in SG/LX environment.

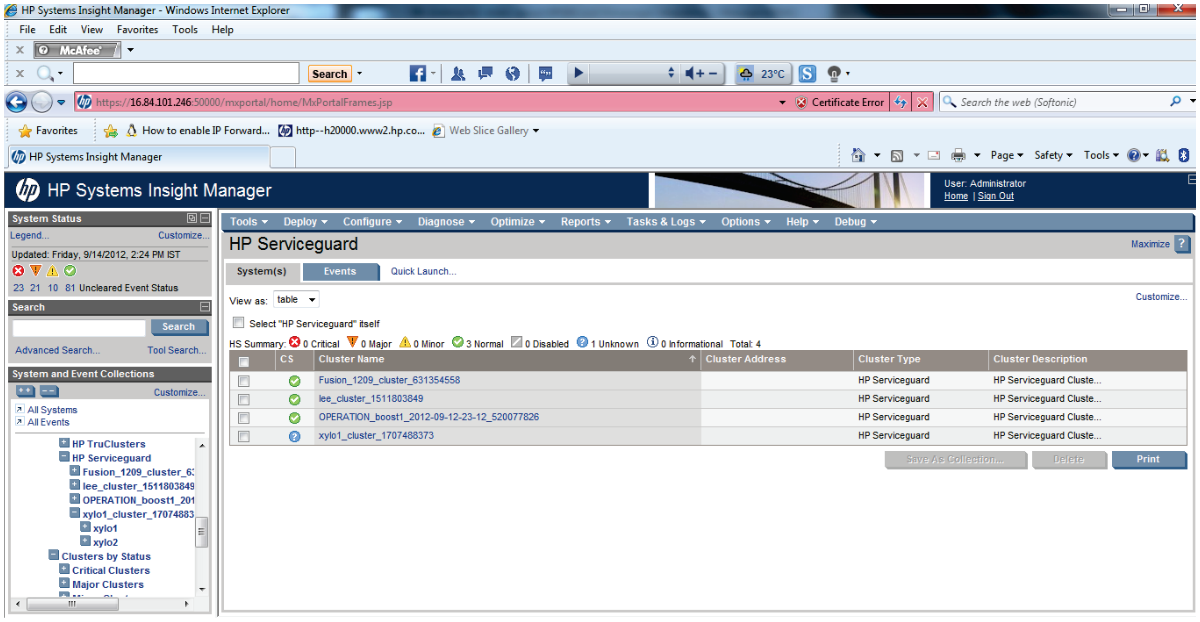
Administering an SG/LX cluster for a VCS administrator

Serviceguard Manager (GUI)

SG/LX cluster can be administered by GUI. The Serviceguard GUI is called Serviceguard Manager which simplifies the configuration and administration of cluster and packages. Serviceguard Manager is a browser-based management tool available as HP System Management Homepage (HP SMH) plug-in. HP Serviceguard Manager plug-in for HP SMH can be launched from any of the nodes of a HP SG/LX cluster with a supported web browser. It is installed and automatically configured as part of SG/LX installation.

The HP System Insight Manager (HP SIM) can be used to manage multiple SG/LX clusters. The HP SIM shows status and events from multiple SG/LX clusters, and you can launch HP Serviceguard Manager plug-in from HP SIM for any specific cluster. Please refer the following diagram to see the HP SMH plug-in launched via HP SIM.

Figure 5. HP SIM showing multiple SG/LX clusters


HP Systems Insight Manager - Windows Internet Explorer
File Edit View Favorites Tools Help
https://16.84.101.246:50000/mxportal/home/MxPortalFrames.jsp
Certificate Error
HP Systems Insight Manager
User: Administrator
Home | Sign Out
System Status
Search
System and Event Collections
All Systems
All Events
HP TruClusters
HP Serviceguard
Clusters by Status
Critical Clusters
Major Clusters
Tools Deploy Configure Diagnose Optimize Reports Tasks & Logs Options Help Debug
Maximize
System(s) Events Quick Launch...
Customize...
Save As Collection... Delete Print

HP Serviceguard

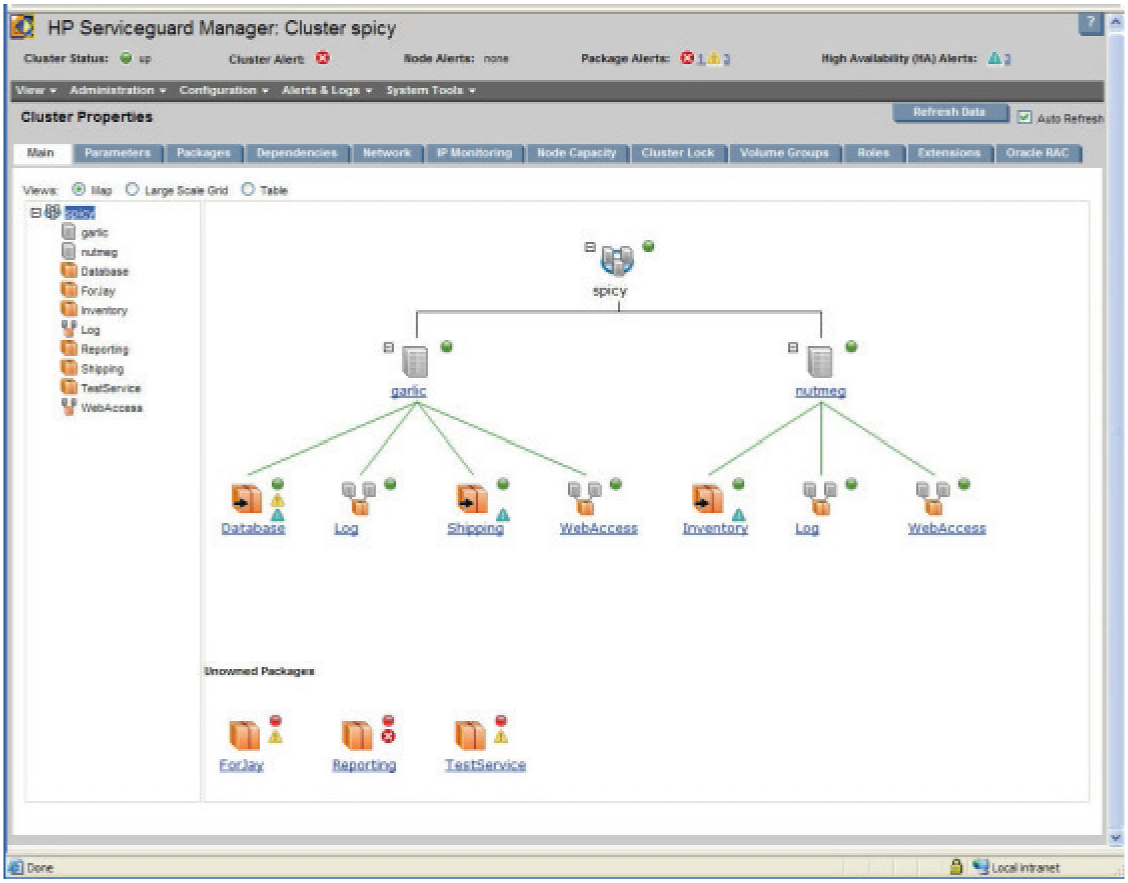
View as: table

□ Select "HP Serviceguard" itself

HS Summary: 0 Critical 0 Major 0 Minor 3 Normal 0 Disabled 1 Unknown 0 Informational Total: 4

□	CS	Cluster Name	Cluster Address	Cluster Type	Cluster Description
□	✓	Fusion_1209_cluster_631354558		HP Serviceguard	HP Serviceguard Cluste...
□	✓	lee_cluster_1511803849		HP Serviceguard	HP Serviceguard Cluste...
□	✓	OPERATION_boost1_2012-09-12-23-12_520077826		HP Serviceguard	HP Serviceguard Cluste...
□	?	xylo1_cluster_1707488373		HP Serviceguard	HP Serviceguard Cluste...

Figure 6. HP SMH Serviceguard Manager plug-in showing a SG/LX cluster



Serviceguard command-line interface

Serviceguard provides rich set of commands to administer cluster and packages. The following table compares some of the important VCS command with SG/LX commands.

Mapping of some important VCS commands to SG/LX commands

Table 5. Some important commands of VCS and SG/LX

	VCS command	Description	Corresponding SG/LX command
1	Hastart	To start the cluster	cmruncl
2	Hastop	To halt the cluster	cmhaltcl
3	hagrp –online <service group> -sys <system>	To bring up a service group/package	cmrunpkg [-n node_name] <package>
4	hagrp –offline <service group> -sys <system>	To halt a service group/package	cmhaltpkg [-n node_name] <package>
5	hastatus –sum	To view the cluster status	Cmviewcl
6	haconf -makerw hagrp -add <service_group> haconf -dump –makero	To add/create service group or package	cmmakepkg <package_configuration_file> fill the package configuration_file cmapplyconf –p <package_configuration_file>

	VCS command	Description	Corresponding SG/LX command
7	haconf -makerw hagrp -modify <service_group> . . . . . haconf -dump –makero	To modify service group or package	cmgetconf –p <package> > <package_configuration_file> Edit the package configuration file as required cmapplyconf –p <package_configuration_file>
6	haconf -makerw hagrp -delete <service_group haconf -dump –makero	To delete service group/package configuration	cmdeleteconf –p <package>
7	haagent -start <agent_name>[-sys]	To start agent/toolkit package	cmrunpkg [-n node_name] <toolkit package>

Migration procedure illustrated with a sample example

This section describes a step-by-step procedure to migrate a VCS cluster running Apache Web server to an SG/LX cluster. The VCS cluster has two systems: hnode and pnode running Red Hat 6.1 and uses three SCSI-3 based coordination point disks as membership arbitration mechanism. There is a single channel bonded heartbeat network (10 .25.1.0) between the nodes.

The Apache Web server application is deployed as a failover service group: “apache_svcgrp” using VCS Apache agent. The Apache Web server is installed and configured locally on both nodes of the cluster (that is, the Apache configuration file, process ID file, and the binary that reside locally on both the nodes of the cluster). The Apache document root (/mount/vg_apache/apache) resides on a shared disk (/dev/vgapache/lvol0) accessible from both the nodes.

Figure 7. Symantec VCS cluster running the highly available Apache Web server

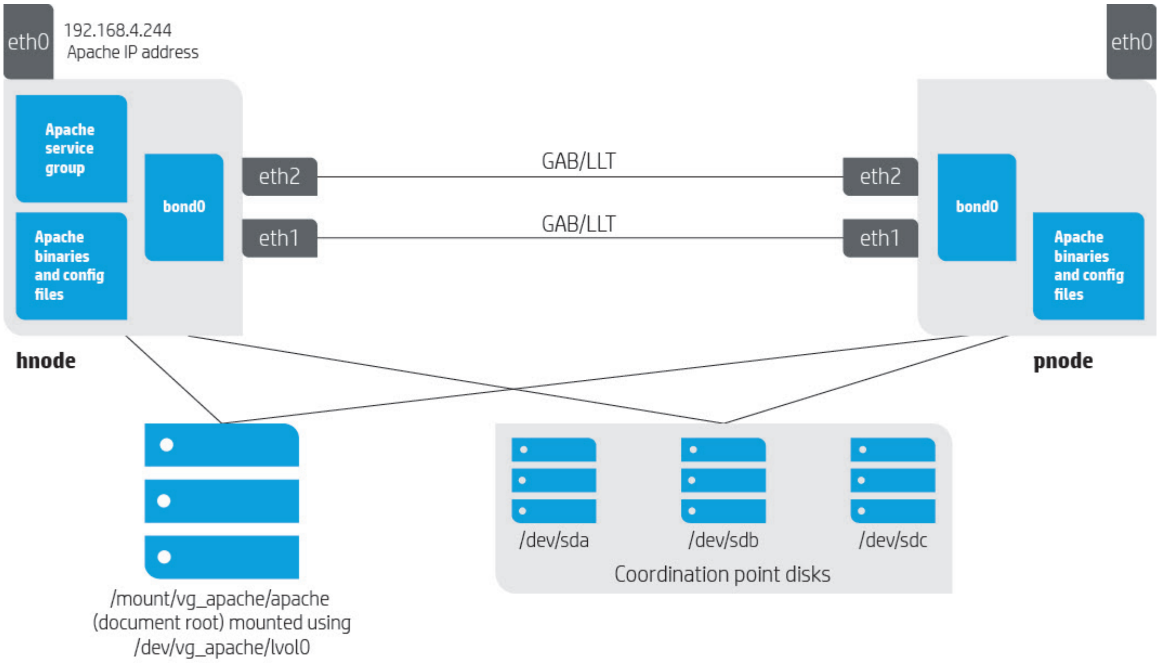
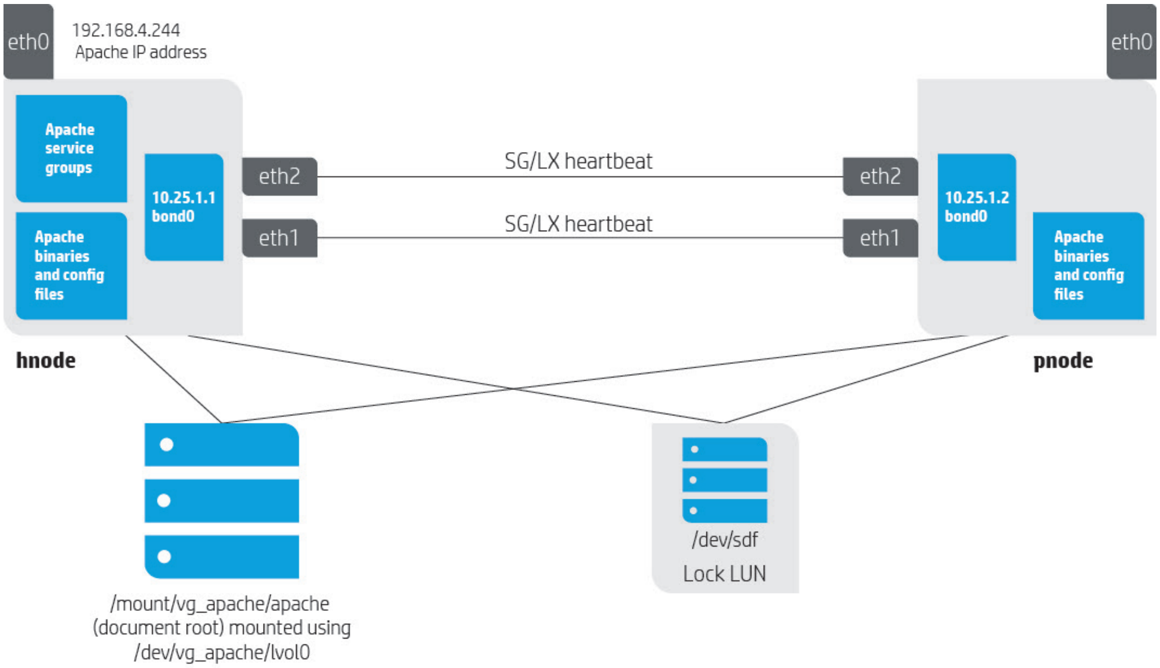


Figure 8. Target SG/LX cluster running the highly available Apache Web server



The migration procedure

Overview

This migration procedure can convert the existing VCS cluster to SG/LX cluster using the same nodes and heartbeat network. The target SG/LX cluster can use a new shared disk as a cluster Lock LUN mechanism for membership arbitration. The Apache web server configured in the service group: “apache_svcgrp” can be converted to an SG/LX package with the name “apache_package” using the SG/LX contributed toolkit.

The sample Apache service group uses the following resource agents in the VCS cluster:

- Apache
- IP
- NIC
- LVM volume group
- LVM logical volume
- Mount
- Disk reservation

In this migration procedure, the same Apache application installation and configuration used in the VCS cluster is being used in the SG/LX cluster. The Apache Web server documents (HTML files) kept in a shared storage is also used as is in the SG/LX package. The Apache application is not reinstalled or reconfigured for this migration.

NOTE:
In case the application data in the VCS cluster is kept in a filesystem or volume manager not supported by SG/LX, you must migrate the application data to a filesystem or volume manager supported by SG/LX.

Following are the steps to migrate a sample Apache service group from a VCS cluster to an SG/LX cluster:

- Plan SG/LX heartbeat network and membership arbitration
- Collect the resources information used by your VCS service group
- Back up the application and configuration data
- Install Serviceguard and Apache toolkit software, and prepare the nodes
- Create shared storage for application data

- Configure the SG/LX cluster
- Stop the VCS cluster
- Start the SG/LX cluster
- Verify the SG/LX cluster
- Create and configure the Serviceguard Apache toolkit package
- Start the Serviceguard Apache toolkit package
- Verify the Apache toolkit package
- Remove the VCS configuration and software

Plan SG/LX heartbeat network and membership arbitration

For SG/LX cluster, it is recommended to configure at least two networks or one channel bonded network for heartbeat. In a migration, you can also configure the network used by your VCS cluster for GAB/LLT as heartbeat network for the target SG/LX cluster. VCS heartbeat network configuration is kept in the /etc/llttab file of each node.

A typical/etc/llttab file looks like below:

```
set-node node 1
   set-cluster 2
    link en1 /dev/dlpi/en:1 - ether - -
    link en2 /dev/dlpi/en:2 - ether - -
```

The third field of the lines starting with the directive “link” represents the network device used for GAB/LLT by VCS. SG/LX heartbeat is at network layer (TCP) opposed to VCS heartbeat at link layer. So you might need to configure an IP subnet or use an existing IP subnet on these network devices for SG/LX heartbeat.

In this illustration, the VCS cluster uses a channel bonded interface for GAB/LLT. The /etc/llttab looks like this:

```
set-node hnode
    set-cluster 26020
    link bond0 bond0 - ether - -
```

The target SG/LX cluster can use an IP network over the “bond0” network interface as heartbeat network. If the “bond0” interface is not configured with any IP network in the cluster nodes, configure an IP network so that SG/LX can detect and use the same for heartbeat.

For membership arbitration in the target SG/LX cluster, you can choose to use a cluster Lock LUN or Quorum Server. When using Quorum Server for membership arbitration, ensure that all the nodes can reach the server on the network.

If you choose to use a cluster Lock LUN, ensure that you use a new disk. Do not use any of the VCS coordination point disks or any other disk in use with the existing VCS cluster as a cluster Lock LUN in SG/LX. You can identify the coordination point disk being used by the VCS in the /etc/vxfentab file.

In this illustration, the /etc/vxfentab can have the following values:

```
/dev/sda
/dev/sdb
/dev/sdc
```

Excluding the above Coordination point disks, we can use a new disk /dev/sdf in the target SG/LX cluster as cluster Lock LUN.

Collect the resources information used by your VCS service group

The resource information of the VCS service group for your application can be used later while creating an SG/LX package for the same application in the SG/LX cluster. The service group resource information typically includes storage (filesystems, volume managers), network (IP subnets), and other OS resources used by the application.

- Collect the following details related to the IP being used by the application:
  - IP device
  - IP address
  - Netmask
  - NIC device
- Collect the following detail related to the volume group being used by the application:
  - Volume group
- Collect the following details related to the logical volume group being used by the application:
  - Logical volume and corresponding volume group
- Collect the following details related to filesystem being used by the application:
  - MountPoint
  - FSType
  - FsckOpt
  - BlockDevice

In addition to the resource information, there can be VCS application agent-specific information in the VCS service group that can also be useful when configuring SG/LX package using an SG/LX toolkit. This typically includes the following:

- Collect the following details related to the application:
  - Application configuration file path
  - Application home or binaries dir
  - Application startup and shutdown behavior flags

In this illustration, the following table has the VCS Apache service group resource information and the VCS Apache agent specific information that can be required while creating an SG/LX package for the Apache Web Server.

Table 6. Package worksheet

Attributes	Values
Service group	apache_svcgrp (You can use same name for SG/LX package)
ConfigFile	/etc/httpd/conf/httpd.conf
IP device	eth0
IP address	192.168.4.244
Netmask	255.255.254.0
NIC device	eth0
Logical volume	/dev/vg_apache/lvol0
Volume group	vg_apache
Mount	/mount/vg_apache/apache
FS_type	ext3

Back up the application and configuration data

You should always back up the application data when you are going to make significant changes to a system configuration. It is recommended to create a backup disk with all the application data.

Install Serviceguard and Apache toolkit software, and prepare the nodes

Install Serviceguard and Apache toolkit software on each node of the cluster. The Apache toolkit software is a part of the Serviceguard Contributed Toolkit which is free of cost. Serviceguard provides a utility “cmeasyinstall” that simplifies the installation process. It installs the Serviceguard components RPMs and list all prerequisite RPMs required for the Serviceguard components RPM installation. Please refer the section “Gather Required Software “of” SG/LX Deployment Guide” (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03393068/c03393068.pdf) to get details about Serviceguard software installation.

Please refer sections “Preparing the Servers”, “Serviceguard System Configuration Automation script”, “Allowing Root Access to an unconfigured Node”, and “Configuring Name Resolution” of “SG/LX Deployment Guide” (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03393068/c03393068.pdf) to get details about preparing nodes for the SG/LX cluster.

The Apache application should not be reinstalled or reconfigured on the nodes. The same Apache application installation and configuration used in the VCS cluster can be used in the SG/LX cluster.

Create the shared storage for application data

As in VCS cluster, the application data that can change must be kept in a shared storage accessible from all the cluster nodes. Application binaries and any application data that is smaller in size and don’t change can be kept locally in each cluster node.

If the filesystem or volume manager used by the application to store its data in the VCS cluster is also supported by SG/LX, then the same storage layout can be used as is in an SG/LX cluster also. If the filesystem or volume manager used by the application to store data in the VCS cluster is not supported with SG/LX, then it is required to create a fresh storage layout for application data using filesystem or volume managers supported by SG/LX.

For more recent list of SG/LX supported filesystems and volume managers please refer to the “HP SG/LX Certification Matrix.” (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03398707/c03398707.pdf)

Configure the SG/LX cluster

CLI

- Execute the following command to create the cluster ASCII file from either of the nodes as root user.

```
cmquerycl -L /dev/sdf –n hnode –n pnode –C cluster.ascii
```

Some important attributes of the cluster.ascii file are listed below:

CLUSTER_NAME	hnode_cluster
NODE_NAME	hnode
NETWORK_INTERFACE	eth0
STATIONARY_IP	192.168.4.230
NETWORK_INTERFACE	bond0
HEARTBEAT_IP	10.25.1.1
CLUSTER_LOCK_LUN	/dev/sdf

NODE_NAME	pnode
NETWORK_INTERFACE	eth0
STATIONARY_IP	192.168.4.231
NETWORK_INTERFACE	bond0
HEARTBEAT_IP	10.25.1.2
CLUSTER_LOCK_LUN	/dev/sdf

MEMBER_TIMEOUT	14000000 (default value)

The cmquerycl command can automatically detect the network topology between the nodes and configure the eligible networks for heartbeat. The command can setup the cluster Lock LUN using the disk passed through –L option. When using a Quorum Server, use the -q <Quorum Server name> option instead.

You can refer to cluster attributes comparison table VCS cluster attributes and SG/LX cluster attributes and section Plan SG/LX heartbeat network and membership arbitration, to modify other SG/LX cluster attributes to match you VCS cluster behavior.

cmquerycl can automatically specify the name of the cluster as the name of the node on which cmquerycl is run with a suffix: "_cluster". In this illustration, the cluster name is specified as "apache_cluster", by modifying the cluster_name attribute in the generated cluster.ascii file.

- Execute the following command to apply the cluster ASCII file as root user:
  – cmapplyconf –C /cluster.ascii

GUI (Serviceguard Manager)

- You can also configure an SG/LX cluster using Serviceguard Manager. Please refer the section "Configure the Cluster" of "HP SG/LX Deployment Guide" (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03393068/c03393068.pdf) and follow the steps mentioned there to configure an SG/LX cluster using Serviceguard Manager. You can refer to cluster attributes comparison table VCS cluster attributes and SG/LX cluster attributes and section Plan SG/LX heartbeat network and membership arbitration to modify other SG/LX cluster attributes to match you VCS cluster behavior.

Stop the VCS cluster

Log in as root user and execute the following commands:

- Bring down the Apache agent service group on the hnode (assuming it was running on the hnode).
  – hagrp -offline apache_agent_servicegroup -sys hnode
- Check the status of Apache agent service group. It should be "offline".
  – hagrp -state apache_agent_servicegroup
- Stop VCS on both nodes (hnode and pnode), and execute the following command on the both nodes.
  – hastop –local
- Verify that the VCS engine port h is closed.
  – gabconfig –a
- Stop I/O fencing if you have configured it. Stop GAB and then LLT using following commands:
  – /etc/init.d/vxfen stop
  – /etc/init.d/gab stop
  – /etc/init.d/llt stop

Start the SG/LX cluster

CLI

Execute the following command to start the cluster:

cmruncl

```
cmruncl: Validating network configuration...
cmruncl: Network validation complete
Waiting for cluster to form.... done
Cluster successfully formed.
Check the syslog files on all nodes in the cluster to verify that no warnings occurred during startup.
```

GUI (Serviceguard Manager)
Launch Serviceguard Manager from one of the node and go to the “Administration” menu and select “Run Cluster”.

Verify the SG/LX cluster

CLI
Execute the following command to view the status of the cluster. The status should be “up”.

cmviewcl

CLUSTER	STATUS
apache_cluster	up

NODE	STATUS	STATE
Hnode	up	running
Pnode	up	running

GUI (Serviceguard Manager)
Launch Serviceguard Manager from one of the nodes and click on the cluster name to view the cluster status. The cluster status should be “up”.

Create and configure the Serviceguard Apache toolkit

CLI
Please follow the following steps to create and configure Serviceguard Apache toolkit:

- Create Apache toolkit package directory.
  - mkdir $SGCONF/pkg/Apache
- Change directory to the Apache toolkit package directory.
  - cd $SGCONF/pkg/Apache
- Create the package configuration template using the cmmakepkg command and include appropriate modules required by your application. In this illustration, we can use the tkit/apache module that is available through the SG/LX Contributed Toolkit product.
  - cmmakepkg -m tkit/apache/apache apache_package.conf

cmmakepkg command, by default, includes all the modules necessary to manage typical application resources like filesystem, volume group, IP subnet, etc., in a package configuration file. In addition, you can select the application specific modules that can start/stop/monitor applications. The application specific modules are available through corresponding toolkit product. To get a list of modules installed in the cluster, use the cmmakepkg –l command. If you do not find an appropriate module for your application buy or get the appropriate SG/LX toolkit for the application and install the same on all nodes.

If there are no toolkits available for your application, you must write your own using SG/LX Developer’s Toolbox that is freely downloadable from software.hp.com.

- Edit the package configuration file and configure the attributes using the VCS resource information and VCS agent information collected at the step “Collect the resources information used by your VCS service group” in Table 6. Package worksheet.

Use the following table to map VCS service group resource attributes to SG/LX package attributes for this illustration:

VCS service group resource attributes	**Corresponding SG/LX package attributes**	**Descriptions**
Service group	package_name	Represent the name of service group/package
ConfigFile	httpd_config_file	Represent the path of Apache Web server configuration file
IP address	ip_address	Represent the IP address used by the application
Logical volume	fs_name	Represent the shared device name which is being used by the application
Volume group	Vg	Represent the volume group name being used by the application
Mount	fs_directory	Represent the location where the filesystem being used by the application is mounted
FS_type	fs_type	Represent the type of the filesystem type being used by the application

Using this approach, the SG/LX package configuration file in this illustration can be populated as below:

SG/LX package attributes	
package_name	apache_package (You can use the same VCS service group name also)
tkit_dir	$SGCONF/pkg/Apache
httpd_config_file	/etc/httpd/conf/httpd.conf
ip_subnet	192.168.0.0
ip_address	192.168.4.244
vg	vg_apache
fs_name	/dev/vg_apache/lvol0
fs_directory	/mount/vg_apache/apache
fs_type	ext3

- Apply the package configuration.
  – cmapplyconf -P apache_package.conf

GUI (Serviceguard Manager)

Please refer the sections “Create Package from Serviceguard Manager” and “Customize and Distribute Toolkit Files” of “HP SG/LX Deployment Guide” (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03393068/c03393068.pdf) to get details on using Serviceguard Manager to configure Apache toolkit package. Please refer to the preceding table while configuring the Apache toolkit package.

Please refer section “Apache Toolkit” of “HP Serviceguard Contributed Toolkit Suite on Linux User Guide” (http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03603547/c03603547.pdf) to get more details of Apache toolkit configuration.

Start the Serviceguard Apache toolkit package

CLI

Execute following command to start the cluster:

```
cmrunpkg apache_package
```

GUI (Serviceguard Manager)

Launch Serviceguard Manager from one of the nodes, Select the “apache_package”, go to the “Administration” menu and select “Run Package”.

Verify the Apache toolkit package

CLI

- Execute the following command to view the status of the cluster:

```
cmviewcl
```

CLUSTER	STATUS
apache_cluster	up

NODE	STATUS	STATE
Hnode	up	running

PACKAGE	STATUS	STATE	AUTO_RUN	NODE
apache_package	up	running	enabled	hnode

NODE	STATUS	STATE
Pnode	up	running

- Test that you can access the apache_package content from your browser at the following URL:
  http://192.168.4.244:80/index.html
- Try halting the package with the following command:
  cmhaltpkg –n hnode apache_package
- Try starting the package on other node with the following command:
  cmrunpkg –n pnode apache_package
- Test that you can access the apache_package content from your browser at the following URL:
  http://192.168.4.244:80/index.html

GUI (Serviceguard Manager)

Please refer the “Verification” section of “HP SG/LX Deployment Guide”
(http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03393068/c03393068.pdf)
to get the details on how to verify the “apache_package” using Serviceguard Manager.

Remove the VCS configuration and software

After verifying the Apache toolkit package, you can remove the VCS configuration and VCS software. Please refer to “Uninstalling VCS” section of “Veritas™ Cluster Server Installation Guide” for Linux.

Troubleshooting tips

Refer to "Troubleshooting Your Cluster" of "Managing HP SG/LX"
(http://bizsupport2.austin.hp.com/bc/docs/support/SupportManual/c03393076/c03393076.pdf)

Terms

FC	Fiber channel
GAB	Group Membership Services and Atomic Broadcast
GUI	Graphical user interface
HA	High availability
HAD	High-availability daemon
LVM	Logical volume management facilities on Linux
LLT	Low Latency Transport
OS	Operating system
SG/LX	Serviceguard for Linux
SPOF	Single point of failure
VCS	Veritas Cluster Server

For more information

- HP SG/LX product documentation at hp.com/go/linux-serviceguard-docs
- HP SG/LX certification matrix showing servers, storage, and software versions supported at hp.com/info/sglx
- Veritas documents at https://sort.symantec.com/documents/doc_details/vcs/6.0/Linux/ProductGuides/
- Download SG/LX evaluation software for FREE from following link:
  https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=SGLX1120-DEMO.
  The Serviceguard for Linux evaluation software can be used only for 90 days.

Learn more at
hp.com/go/sglx

Sign up for updates
hp.com/go/getupdated

★ Rate this document